What is claimed is:

1. A method comprising:

feeding back high order bits of a previously executed arithmetic instruction, generated by a first plurality of arithmetic structures, to a second plurality of arithmetic structures; and

using the second arithmetic structures, generating a first partial result of a currently executed arithmetic instruction, the first partial result representing the high order bits summed with low order bits of a result of a first number multiplied by a second number, the summing of the high order bits being performed during multiplication of the first number and the second number, the summing and at least a portion of the multiplication being performed in the second arithmetic structures.

2. The method as recited in claim 1 wherein the high order bits are fed back in redundant number representation.

3. The method as recited in claim 2 wherein the redundant number representation includes sum and carry bits.

4. The method as recited in claim 1 further comprising feeding back the high order bits through a register to the second arithmetic structures.

5. The method as recited in claim 1

generating a second partial result of the currently executed arithmetic instruction in the first arithmetic structures, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number.

6. The method as recited in claim 1

generating a second partial result of the currently executed arithmetic instruction, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second

number summed with the high order bits of the previously executed arithmetic instruction.

7. The method as recited in claim 6 further comprising supplying values generated in one or more most significant columns of the second arithmetic structures to one or more least significant columns of the first arithmetic structures while generating the first and second partial results.

8. The method as recited in claim 5 wherein the generating of the first and second partial result is in response to execution of a single arithmetic instruction.

9. The method as recited in claim 6 wherein the generating of the first and second partial result is in response to execution of a single arithmetic instruction.

10. The method as recited in claim 1 wherein the arithmetic structures are comprised of carry save adder tree columns.

11. The method as recited in claim 1 wherein the arithmetic structures are comprised of Wallace tree columns.

12. The method as recited in claim 1 wherein the arithmetic structures are utilized in performing both integer and XOR multiplication.

13. The method as recited in claim 12 further comprising using a logical circuit in the first and second plurality of arithmetic structures to supply a fixed value in XOR multiplication mode and a variable value for integer multiplication mode that varies according to inputs supplied to the logical circuit, to thereby ensure a result is determined in XOR multiplication unaffected by carry logic performing carries in integer multiplication mode.

14. The method as recited in claim 13 wherein the logical circuit operates as a majority circuit in integer multiplication mode and outputs a zero in the XOR multiplication mode.

15. The method as recited in claim 1 wherein the first partial result is in redundant number representation.

16. The method as recited in claim 15 further comprising supplying the first partial result to an adder circuit to generate a non redundant representation of the first partial result and a carry out value.

17. The method as recited in claim 16 further comprising feeding back the carry out value to the adder circuit.

18. The method as recited in claim 16 method further comprising feeding back the carry out value to the second plurality of arithmetic structures.

19. The method as recited in claim 1 further comprising feeding back high order bits of the current arithmetic instruction from the first arithmetic structures to the second arithmetic structures for use with execution of a subsequent single arithmetic instruction.

20. The method as recited claim 1 further comprising storing the high order bits into an extended carry register.

21. A method comprising:

feeding back high order bits of a previously executed arithmetic instruction, from a first plurality of arithmetic structures generating the high order bits, to a second plurality of arithmetic structures;

supplying a third number to the second plurality of arithmetic structures; and

using the second arithmetic structures generating a first partial result of a currently executed arithmetic instruction, the first partial result being a representation of the high order bits summed with, low order bits of a result of a first number multiplied by a second number, and summed with the third number, the summing of the high order bits and the summing of the third number being performed during multiplication of the first number and the second number, the summing and a portion of the multiplication being performed in the second arithmetic structures.

22. The method as recited in claim 21 further comprising feeding back the high order bits through a register to the second arithmetic structures.

23. The method as recited in claim 21 further comprising:

generating a second partial result of the currently executed arithmetic instruction in the first arithmetic structures, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number.

24. The method as recited in claim 21 further comprising:

generating a second partial result of the currently executed arithmetic instruction, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number summed with the high order bits of the previously executed arithmetic instruction and the third number.

25. The method as recited in claim 24 further comprising supplying values generated in one or more most significant columns of the second arithmetic structures to one or more least significant columns of the first arithmetic structures while generating the first and second partial results.

26. The method as recited in claim 23 wherein the generating of the first and second partial result is in response to execution of a single arithmetic instruction.

27. The method as recited in claim 21

supplying the first partial result to an adder circuit to generate a non redundant representation of the first partial result and a carry out value.

28. The method as recited in claim 27 further comprising feeding back the carry out value to the adder circuit.

29. The method as recited in claim 27 method further comprising feeding back the carry out value to the second arithmetic structures.

30. The method as recited in claim 21 wherein the arithmetic structures are comprised of Wallace tree columns.

31. The method as recited in claim 21 wherein the arithmetic structures are comprised of carry save adder tree columns.

32. The method as recited in claim 21 wherein the arithmetic structures are utilized in performing both integer and XOR multiplication.

33. The method as recited in claim 32 further comprising using a logic circuit in the first and second plurality of arithmetic structures to supply a fixed value in XOR multiplication mode and a variable value for integer multiplication mode that varies according to inputs supplied to the logical circuit, to thereby ensure a result is determined in XOR multiplication unaffected by carry logic performing carries in integer multiplication mode.

34. The method as recited in claim 33 wherein the logic circuit operates as a majority circuit in integer multiplication mode and outputs a zero in the XOR multiplication mode.

35. The method as recited in claim 21 wherein the high order bits are in redundant number representation.

36. The method as recited in claim 21 further comprising feeding back high order bits of the current arithmetic instruction from the first arithmetic structures to the second arithmetic structures for use with execution of a subsequent single arithmetic instruction.

37. The method as recited in claim 21 further comprising storing the high order bits into an extended carry register.

38. An apparatus comprising:

a first plurality of arithmetic structures generating high order bits for an arithmetic operation that includes a multiplication operation;

a second plurality of arithmetic structures generating low order bits of the arithmetic operation; and

wherein the second arithmetic structures are coupled to receive the high order bits generated by the first plurality of arithmetic structures during a previous arithmetic operation and to generate a first partial result of the arithmetic operation, the first partial result representing the high order bits summed with low order bits of a multiplication result of the multiplication operation.

39. The apparatus as recited in claim 38 wherein a second partial result of the arithmetic instruction is generated in the first arithmetic structures, the second partial result representing the high order bits of the arithmetic operation.

40. The apparatus as recited in claim 39 further wherein values generated in one or more most significant columns of the second arithmetic structures are supplied to one or more least significant columns of the first arithmetic structures while generating the first and second partial results.

41. The apparatus as recited in claim 39 wherein the generating of the first and second partial results is in response to execution of a single arithmetic instruction.

42. The apparatus as recited in claim 38 further comprising a register coupled to the first and second arithmetic structures to supply the high order bits to the second arithmetic structures.

43. The apparatus as recited in claim 38 wherein the first partial result is in redundant number representation.

44. The apparatus as recited in claim 43 further comprising an adder circuit coupled to receive the first partial result and to generate a non redundant representation of the first partial result and a carry out value.

45. The apparatus as recited in claim 44 wherein the carry out value is fed back to the adder circuit.

46. The apparatus as recited in claim 44 method wherein the carry out value is fed back to the second arithmetic structures.

47. The apparatus as recited in claim 38 wherein the arithmetic structures are comprised of Wallace tree columns.

48. The apparatus as recited in claim 38 wherein the arithmetic structures are comprised of carry save adder tree columns.

49. The apparatus as recited in claim 38 wherein the arithmetic structures are configured to selectively perform one of integer and XOR multiplication according to a control signal.

50. The apparatus as recited in claim 49 further comprising a plurality of logic circuits in the first and second plurality of arithmetic structures, each logic circuit responsive to the control signal to supply a fixed output value in XOR multiplication mode and a variable output value in integer multiplication mode, the variable output value varying according to values of inputs supplied to the logic circuit, to thereby ensure a result is determined in XOR multiplication mode unaffected by carry logic generating carries in integer multiplication mode.

51. The apparatus as recited in claim 50 wherein the logical circuit operates as a majority circuit in integer multiplication mode and outputs a zero in the XOR multiplication mode.

52. The apparatus as recited in claim 38 wherein the apparatus is a general purpose computer.

53. An apparatus comprising:

a first plurality of arithmetic structures generating high order bits for an arithmetic operation that includes a multiplication operation of a first and a second number;

a second plurality of arithmetic structures generating low order bits of the arithmetic operation; and

wherein the second arithmetic structures are coupled to receive the high order bits generated by the first plurality of arithmetic structures during a previous arithmetic operation and are coupled to receive a third number and are coupled to generate a first partial result of the arithmetic operation, the first partial result representing the high order bits summed with, low order bits of a multiplication result of the multiplication operation, and summed with the third number.

54. The apparatus as recited in claim 53 wherein a second partial result of the arithmetic instruction is generated in the first arithmetic structures, the second partial result representing the high order bits of the arithmetic operation.

55. The apparatus as recited in claim 54 further wherein values generated in one or more most significant columns of the second arithmetic structures are supplied to one or more least significant columns of the first arithmetic structures while generating the first and second partial results.

56. The apparatus as recited in claim 54 wherein the generating of the first and second partial result is in response to execution of a single arithmetic instruction.

57. The apparatus as recited in claim 53 further comprising a register coupled to the first and second arithmetic structures to supply the high order bits to the second arithmetic structures.

58. The apparatus as recited in claim 53 further comprising an adder circuit coupled to receive the first partial result and to generate a non redundant representation of the first partial result and a carry out value.

59. The apparatus as recited in claim 58 wherein the carry out value is fed back to the adder circuit.

60. The apparatus as recited in claim 58 method wherein the carry out value is fed back to the second arithmetic structures.

61. The apparatus as recited in claim 53 wherein the arithmetic structures are Wallace tree columns.

62. The apparatus as recited in claim 53 wherein the arithmetic structures are comprised of carry save adder tree columns.

63. The apparatus as recited in claim 53 wherein the arithmetic structures are configured to selectively perform one of integer and XOR multiplication according to a control signal.

64. The apparatus as recited in claim 63 further comprising a plurality of logic circuits in the first and second plurality of arithmetic structures, each logic circuit responsive to the control signal to supply a fixed output value in XOR multiplication mode and a variable output value in integer multiplication mode, the variable output value varying according to values of inputs supplied to the logic circuit, to thereby ensure a result is determined in XOR multiplication mode unaffected by carry logic generating carries in integer multiplication mode.

65. The apparatus as recited in claim 64 wherein the logical circuit operates as a majority circuit in integer multiplication mode and outputs a zero in the XOR multiplication mode.

66. An apparatus comprising:

means for feeding back high order bits of a previously executed arithmetic instruction, generated by a first plurality of arithmetic structures, to a second plurality of arithmetic structures generating low order bits of a currently executed arithmetic instruction; and

means for using the second arithmetic structures to generate a first partial result of the currently executed arithmetic instruction, the first partial result representing the high order bits of the previously executed arithmetic instruction that are summed with low order bits of a multiplication result of a first number multiplied by a second number.

67. An apparatus comprising:

means for feeding back high order bits of a previously executed arithmetic instruction, from a first plurality of arithmetic structures generating the high order bits, to a second plurality of arithmetic structures generating low order bits of a currently executed arithmetic instruction;

means for supplying a third number to the second plurality of arithmetic structures; and

means for using the second arithmetic structures to generate a first partial result, the first partial result being a representation of the high order bits of the previously executed arithmetic instruction summed with low order bits of a result of a first number multiplied by a second number, and summed with the third number.